

Perancangan dan Pengujian Piranti Pemantauan Visual untuk Menentukan Volume Lalu Lintas

Alfina¹, Yuwaldi Away¹, dan Muhammad Isya²

¹Program Studi Magister Teknik Elektro, Fakultas Teknik, Universitas Syiah Kuala

²Program Studi Magister Teknik Sipil, Fakultas Teknik, Universitas Syiah Kuala

Jl. Tengku Syech Abdul Rauf, No. 7, Darussalam, Banda Aceh 23111

e-mail: lovefina4@gmail.com

Abstrak— Sistem transportasi pintar atau *Intelligent Transportation System (ITS)* menerapkan teknologi informasi dan komunikasi untuk pengelolaan transportasi. Permasalahan pada penelitian adalah perlunya piranti visi komputer untuk menentukan *volume* lalu lintas di jalan raya. Perancangan piranti visi komputer ini dilakukan menggunakan kamera statis dan metode *bounding box* untuk menentukan volume lalu lintas berdasarkan jenis kendaraan yaitu kendaraan ringan, kendaraan berat (truk), dan sepeda motor yang diterapkan pada *Gaussian Mixture Models (GMM)*. Pengklasifikasian kendaraan dilakukan berdasarkan ukuran luasan piksel dari objek kendaraan. Pengujian dilakukan di Jalan T. Nyak Arief, Jambotape, Banda Aceh. Dari hasil pengujian diperoleh tingkat ketelitian sistem sebesar 79,32 %.

Kata kunci: *intelligent transportation system, computer vision, counting vehicle, matlab, bounding box*

Abstract—*Intelligent Transportation System (ITS)* applies information and communication technologies for transportation management. Problem in the study is the importance of computer vision tools to determine the volume of traffic on the highway. The design of computer vision device is performed using a static camera and the bounding box method for determining the traffic volume based on vehicle type, namely light vehicles, heavy vehicles (trucks) and motorcycles that are applied on the Gaussian Mixture Models (GMM). The classification is based on pixel area of vehicle objects. The tests were carried out at Jalan T. Nyak Arief, Jambotape, Banda Aceh. The test results of the system achieved accuracy of 79.32%.

Keywords: *intelligent transportation system, computer vision, counting vehicle, matlab, bounding box*

I. PENDAHULUAN

Data volume lalu lintas merupakan komponen yang sangat diperlukan bagi perencanaan suatu kondisi lalu lintas. Biasanya angka volume lalu lintas selalu berubah dan memerlukan pembaharuan data. Jika pengambilan data dilakukan dengan kamera dan kemudian diproses maka akan diperoleh data volume lalu lintas pada saat itu juga [1].

Pertumbuhan jumlah kendaraan bermotor memberikan dampak pada lingkungan, seperti kemacetan, kecelakaan dan pencemaran lingkungan. Dari dampak tersebut maka diperlukan konsep baru yaitu *Intelligent Transportation System (ITS)*[2]. Sehingga diperlukan perancangan piranti visi komputer untuk menentukan volume lalu lintas

Salah satu pengolahan citra dalam bentuk video adalah untuk melakukan proses pengamatan suatu objek tidak perlu dilakukan terus menerus, namun cukup meletakkan suatu kamera yang mengarah pada objek yang diamati [4].

Dari hasil penelusuran literatur, saat ini diketahui sudah ada beberapa metode yang dapat digunakan untuk menghitung jumlah dan jenis kendaraan. Metode tersebut adalah Metode Visi Komputer Haugh Transform (HT) dan Connected Component Labeling (CCL) [3], Metode

GMM dan Fuzzy Cluster Means (FCM) [9], Metode Fuzzy C-Means dan Segmentasi Deteksi Tepi Canny [10]. Perancangan piranti visi komputer ini dilakukan menggunakan kamera statis dan metode *bounding box* untuk menentukan volume lalu lintas berdasarkan jenis kendaraan yaitu kendaraan ringan, kendaraan berat (truk), dan sepeda motor yang diterapkan pada *Gaussian Mixture Models (GMM)*. Metode ini berbeda dengan penelitian sebelumnya, karena pengklasifikasian kendaraan dilakukan menggunakan *bounding box* berdasarkan ukuran piksel objek kendaraan yaitu dilakukan proses kalibrasi terlebih dahulu, sehingga didapatkan *range* luas piksel masing-masing untuk tiga jenis kendaraan tersebut. Lokasi pengujian dilakukan di Jalan T. Nyak Arief Jambotape Banda Aceh.

II. STUDI PUSTAKA

A. Konsep Intelligent Transportation System

Intelligent Transportation System (ITS) didefinisikan sebagai sebuah sistem yang meningkatkan kualitas transportasi melalui peningkatan keselamatan dan efisiensi pergerakan orang, barang dan informasi dengan mobilitas

yang lebih baik, efisiensi bahan bakar dan pengurangan polusi serta peningkatan efisiensi operasi [5].

B. Segmentasi Citra

Suatu citra menjadi beberapa daerah atau objek yang dimilikinya dengan cara melakukan segmentasi terhadap objek yang ada dalam citra tersebut. Segmentasi citra merupakan suatu proses memecah suatu citra menjadi banyak segmen atau bagian daerah yang tidak saling bertabrakan (*non-overlapping*) [6].

Dalam citra digital daerah hasil segmentasi tersebut merupakan kelompok pixel yang bertetangga atau berhubungan. Segmentasi dilakukan melalui beberapa pendekatan, yaitu [9] :

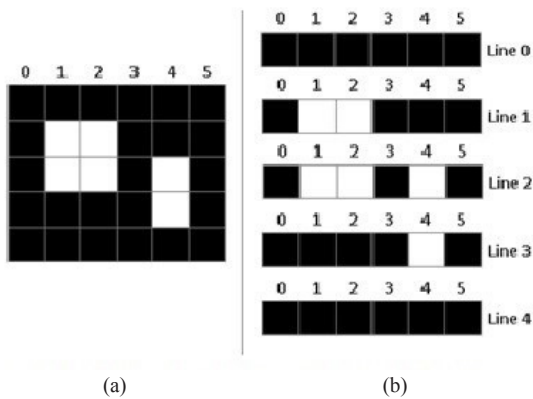
1. Pendekatan batas (*boundary approach*)
2. Pendekatan tepi (*edge approach*)
3. Pendekatan daerah (*region approach*)

Proses segmentasi digunakan dalam berbagai penerapan, meskipun metode yang digunakan bervariasi, namun memiliki tujuan yang sama yaitu mendapatkan representasi sederhana yang berguna dari suatu citra [6].

C. Blob Detection

Dalam *blob detection*, algoritma yang dipakai adalah algoritma *growing regions*. Algoritma ini digunakan untuk menemukan *blob* dalam gambar dan bisa diaplikasikan pada *sequence image*. Contohnya, jika citra yang diaplikasikan adalah citra *grayscale*, maka setiap piksel dari citra tersebut pastilah memiliki nilai yang mengindikasikan tingkat kecerahan dari citra tersebut pada titik piksel tersebut [9]. Dalam algoritma *growing regions* ini, *image* yang dipakai adalah dalam format *binary*. *Frame* yang diolah dengan algoritma ini adalah *frame* asli dan *frame* hasil segmentasi yang dalam tipe *binary*. Algoritma *growing regions* tersebut ditunjukkan pada Gambar 1.

Gambar 1 adalah contoh penerapan algoritma *growing regions* pada *image* dengan ukuran 6x5 piksel. Gambar 1(a) adalah keseluruhan *image* yang telah di segmentasi (warna hitam merupakan *background* dan warna putih merupakan *foreground*). Gambar 1(b) adalah proses



Gambar 1. Pemeriksaan blob per garis dari suatu image

penerapan algoritma *growing regions*.

D. Gaussian Mixture Model (GMM)

GMM adalah sebuah tipe density model yang terdiri dari komponen fungsi-fungsi Gaussian. Komponen fungsi ini terdiri dari Threshold yang berbeda untuk menghasilkan multi *model-model density* dikelompokkan berdasarkan distribusi yang dianggap paling efektif sebagai model latar belakang [9].

Setiap piksel memiliki GMM-nya sendiri dan data yang diolah adalah warna piksel yang didapat dari input. Model-model GMM terbentuk dari data warna piksel berdasarkan waktu [9]. Model yang terbentuk dibagi menjadi 2 bagian, *model background* dan *non background*. Terdapat beberapa tahapan dalam pemilihan distribusi *background* yaitu tahap pencocokkan input terhadap distribusi, dan tahap pemilihan distribusi yang mencerminkan *background* yang berada di atas rasio. Selain itu akan dianggap sebagai *foreground*. Tahapan tahapan tersebut dijelaskan sebagai berikut:

- a. Pencocokkan input terhadap Distribusi Pada tahap ini input dicocokkan dengan semua distribusi sampai ditemukan distribusi yang paling cocok. Suatu piksel dikatakan masuk dalam suatu distribusi jika nilai piksel tersebut masuk dalam jarak 2.5 standar deviasi dari sebuah distribusi. Untuk pencocokkan input digunakan rumus:

$$\mu_k - 2.5 * \sigma_k < X_t < \mu_k + 2.5 * \mu_k \quad (1)$$

Dengan adalah vector dari warna piksel *Red Green Blue* (RGB) pada waktu t , adalah vector nilai mean (RGB) dari Gaussian ke k^{th} , dan sebagai standar deviasi dari Gaussian ke k^{th} [9].

- b. Update Parameter Pada tahap ini dilakukan update terhadap nilai dari parameter-parameter GMM yang nantinya digunakan untuk mengolah input selanjutnya. Nilai yang di-update terdiri dari *weight*, *mean*, dan *varian*. Nilai *weight* di-update tiap waktu. Untuk meng update nilai weight digunakan rumus [9]:

$$\omega_{k,t} = (1 - \alpha) \omega_{k,t-1} + \alpha (M_{k,t}) \quad (2)$$

Dengan adalah weight dari Gaussian ke k^{th} pada waktu t , α adalah *learning rate* dan nilai adalah 1 untuk model yang cocok dan 0 untuk model lainnya. Setelah nilai *weight* diupdate dilakukan normalisasi sehingga total weight dari semua distribusi tidak lebih dari 1. Nilai standar deviasi dari suatu distribusi di-update setiap ada nilai piksel yang cocok dengan distribusi tersebut. Untuk meng update nilai standar deviasi digunakan rumus[9]:

$$\sigma_t^2 = (1 - \rho) \sigma_{t-1}^2 + \rho (X_t - \mu_t)^T (X_t - \mu_t) \quad (3)$$

- c. Pemilihan Distribusi Background Pada tahap ini dipilih model-model yang mencerminkan *background*. Pertama model-model diurutkan berdasarkan ω/σ^2 sehingga distribusi yang paling mencerminkan *background* tetap di atas dan yang tidak mencerminkan *background* ada di bawah yang nantinya digantikan oleh distribusi yang lain. Untuk memilih B distribusi pertama yang dijadikan distribusi *background* digunakan rumus:

$$B = \arg \min_b \sum_{k=1}^b W_k \quad (4)$$

Dengan T adalah batas skala dengan proporsi terkecil dari data yang sebaiknya dihitung sebagai *background* [9].

E. Filter Objek

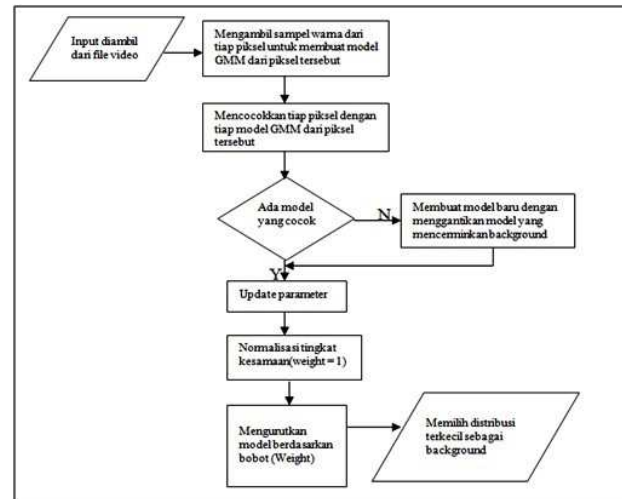
Setiap nilai dari piksel adalah angka yang menunjukkan intensitas atau warna dari gambar. GMM adalah teknik untuk memodelkan warna-warna *background* dari setiap piksel. Tiap piksel memiliki GMM-nya sendiri dan data yang diolah adalah warna piksel yang didapat dari input. Model-model GMM terbentuk dari data warna piksel berdasarkan waktu. Model yang terbentuk dibagi menjadi 2 bagian, model *background* dan model *nonbackground*. Model *background* adalah model yang mencerminkan *background* [9].

Proses yang dilakukan adalah membedakan antara distribusi *Background* dengan distribusi *Foreground*. Proses untuk mengenali distribusi *foreground* adalah membandingkan obyek frame video ke model *background*, untuk menentukan apakah setiap piksel merupakan bagian dari distribusi *Background* atau *Foreground*. Pada perencanaan sistem filter menggunakan GMM terdapat beberapa proses untuk pemilihan distribusi yang mencerminkan *Background* atau tidak. Distribusi yang tidak cocok disebut juga dengan distribusi *Foreground*. Skema proses pemilihan distribusi *Background* menggunakan algoritma GMM tampak pada Gambar 2 [9].

F. Ekstraksi Kendaraan

Ekstraksi Kendaraan dilakukan adalah membedakan antara distribusi *Background* dengan distribusi *Foreground*. Proses untuk mengenali distribusi *foreground* adalah membandingkan obyek frame video ke model *background*, untuk menentukan apakah setiap piksel merupakan bagian dari distribusi *background* atau *Foreground*. Pada perencanaan sistem filter dengan menggunakan GMM terdapat beberapa proses untuk pemilihan distribusi yang mencerminkan *Background* atau tidak. Distribusi yang tidak cocok disebut juga dengan distribusi *Foreground* [9].

Proses ekstraksi kendaraan untuk mencari fitur suatu kendaraan dibutuhkan contoh gambar yang digunakan sebagai acuan. Gambar yang digunakan sebagai acuan



Gambar 2. Skema proses pemilihan distribusi background

untuk mendeskripsikan fitur kendaraan seperti panjang dan lebar kendaraan. Variabel-variabel yang dibutuhkan untuk mendapatkan suatu ciri kendaraan adalah menghitung banyak piksel berdasarkan pada sumbu X dan Y [9].

Gambar 3 menunjukkan bahwa pada sumbu X1 mengindikasikan dari lebar suatu kendaraan, letak posisi titik berada pada titik 4–5 sehingga $X1=2$. Untuk mendapatkan panjang kendaraan mengacu pada sumbu Y. Untuk Y1 mengindikasikan dari panjang suatu kendaraan dimana posisi titik berada pada titik 2–4 sehingga $Y1 = 3$. Sehingga didapat suatu fitur kendaraan berupa panjang kendaraan ($Y1=2$) dan lebar kendaraan ($X1=3$).

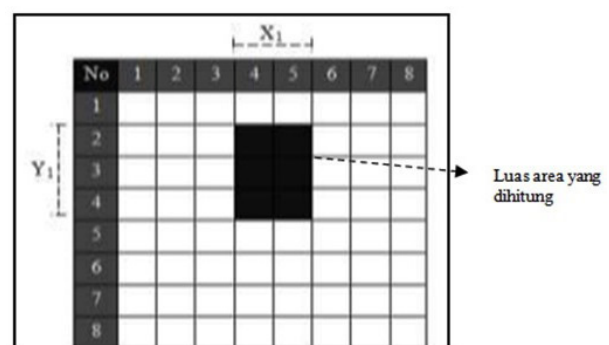
G. Volume Lalu Lintas

Volume lalu lintas adalah jumlah kendaraan yang melewati suatu pias jalan pada jalur gerak untuk satu satuan waktu tertentu, dan biasanya diukur dalam unit satuan kendaraan per satuan waktu. Hal ini dinyatakan dengan rumus [7]:

$$q = \frac{n}{T} \quad (5)$$

Dengan :

q = Volume lalu lintas yang melewati suatu pias tertentu (kendaraan/jam)



Gambar 3. Proses ekstraksi ciri kendaraan Tahapan penelitian

n = Jumlah kendaraan yang melewati pias tersebut dalam interval waktu T (kendaraan)

T = Interval waktu pengamatan (jam)

Volume lalu lintas tidak merata sepanjang waktu dalam sehari, sebulan atau setahun. Dikenal adanya waktu-waktu sibuk dan waktu-waktu sepi. Saatnya waktu-waktu sibuk dan waktu-waktu sepi tersebut tidak sama untuk semua ruas jalan. Hubungannya antara besar volume dan waktu (fluktuasi) tersebut pada letak dan fungsi jalan. Dapat dibedakan antara jalan dalam kota, jalan luar kota, jalan yang melayani lokasi industri, perumahan, perkantoran, rekreasi dan sebagainya [8].

III. METODE

A. Metode Penelitian

Objek penelitian ini adalah perancangan piranti pemantauan lalu lintas untuk menentukan volume lalu lintas dan jenis kendaraan. Adapun tahapan yang digunakan pada penelitian ini tersaji pada Gambar 4.

Penelitian ini menggunakan metode kuantitatif yaitu merupakan suatu studi penggunaan ITS menentukan volume lalu lintas. Perancangan prototipe dimulai dengan menginput kendaraan secara visual melalui kamera dan output dari penelitian ini adalah didapatkan jumlah kendaraan berdasarkan jenis kendaraan.

Data-data yang dikumpulkan adalah jumlah kendaraan yang melewati jalan raya, lebar jalan, lebar bahu Jalan, panjang jalan, dan median jalan.

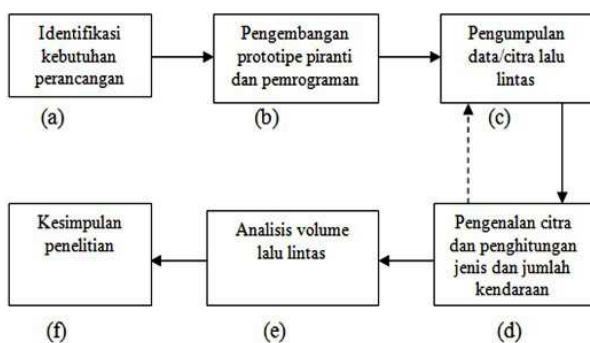
B. Bahan Penelitian

Penelitian ini menggunakan:

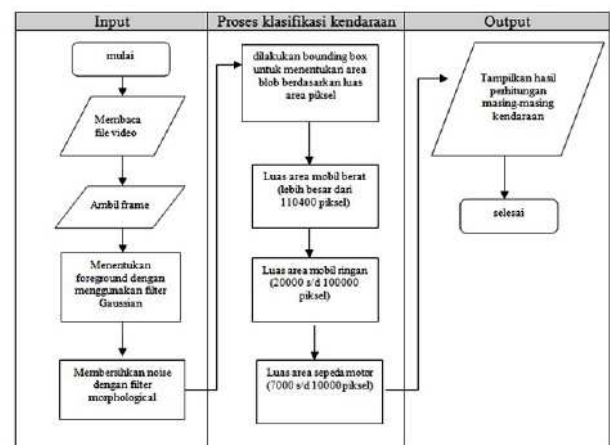
1. kamera digital
2. komputer dengan spesifikasi cukup untuk menjalankan *software* Matlab

C. Prosedur Pengujian

Dalam penelitian menggunakan kamera dengan posisi kamera diam (*static camera*), dan berfokus pada pendeteksian dan menghitung jumlah berdasarkan jenis kendaraan. Gambar 5 adalah proses rancangan sistem



Gambar 4. Tahapan penelitian



Gambar 5. Proses perancangan sistem

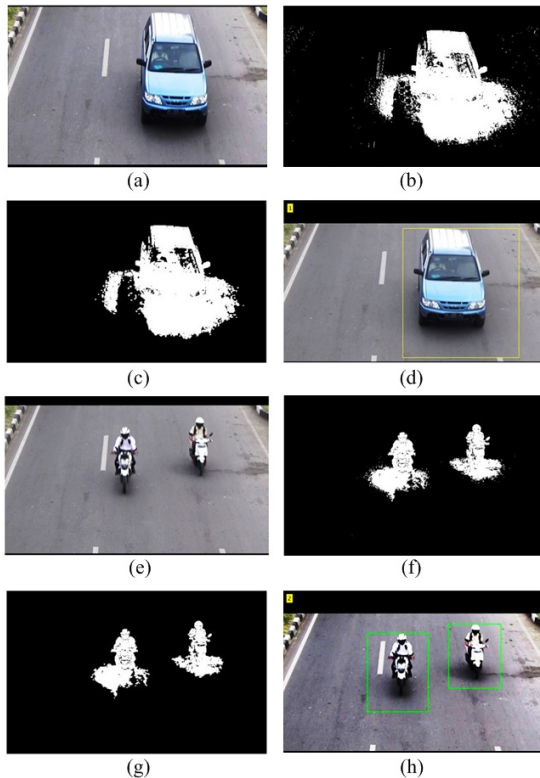
dalam penelitian ini.

Pada Gambar 5 adalah proses perancangan sistem, yaitu terdapat input, proses klasifikasi kendaraan dan output. Pada proses input pertama yang dilakukan adalah membaca file video, dan menginisialisasi video *frame* yaitu bagian *foreground* yaitu objek-objek yang bergerak, dalam hal ini kendaraan, akan disegmentasi dari bagian *background*nya. Pendeteksian *foreground* membutuhkan beberapa jumlah *frame* tertentu untuk menginisialisasikan *Gaussian Mixture Model*. Hasil dari segmentasi *foreground* pada video *frame* ditunjukkan pada Gambar 6(b) dan Gambar 6(f). Pada Gambar 6(b) menunjukkan dari proses segmentasi *foreground*. Pada gambar tersebut dapat dilihat bahwa masih terdapat beberapa gangguan (*noise*) pada beberapa area dalam citra. Pada Gambar 6(g) menunjukkan dari proses segmentasi *foreground* sepeda motor. Pada Gambar 6(g) menunjukkan ada 2 objek yang terpisah sehingga dapat dikenali sebagai 2 objek sepeda motor.

Proses klasifikasi jenis kendaraan dikelompokkan ke dalam 3 (tiga) jenis kendaraan yaitu sepeda motor, kendaraan ringan, kendaraan berat (truk). Pengklasifikasian dilakukan menggunakan blob analisis yaitu teknik membaca ukuran objek berdasarkan luas piksel yang tertera pada layar, ukuran piksel objek akan sangat variatif tergantung dari besar *zoom* pada kamera yang mengakibatkan kendaraan terlihat semakin besar atau semakin kecil pada layar, serta besar ukuran sepeda motor, kendaraan ringan, maupun kendaraan berat yang sebenarnya di lapangan.

Untuk kebutuhan itu maka ukuran tiga jenis kendaraan ini harus didefinisikan ke dalam bahasa pemrograman melalui proses kalibrasi terlebih dahulu, sehingga didapatkan *range* luas piksel yang akurat dan berkesesuaian. Kalibrasi pembacaan luas piksel pada program ini dilakukan melalui cara *trial and error*. Luas area piksel untuk masing-masing jenis kendaraan, adalah sepeda motor 7000 hingga 10000 piksel, kendaraan ringan 20000 hingga 100000 piksel, dan kendaraan berat lebih besar dari 110400 piksel.

Pada Gambar 6(a) adalah salah satu sampel citra asli di jalan raya berupa kendaraan ringan, sebelum dilakukan

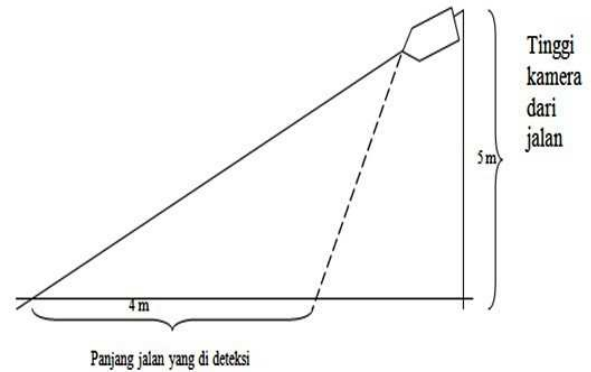


Gambar 6. (a). Citra asli pada kendaraan ringan; (b). Segmentasi foreground kendaraan ringan; (c). Clean foreground pada kendaraan ringan; (d). Blob analisis pada kendaraan ringan; (e). Citra asli sepeda motor; (f). Segmentasi foreground sepeda motor; (g). Clean foreground sepeda motor; (h). Blob analisis sepeda motor

proses segmentasi *foreground* dan *clean foreground*. Pada Gambar 6(c) adalah kualitas citra yang lebih baik dari proses pendeteksian *foreground* dengan menambahkan proses morfologi sebagai *removal noise*. Pada Gambar 6(e) adalah salah satu sampel citra asli di jalan raya berupa kendaraan sepeda motor, sebelum dilakukan proses segmentasi *foreground* dan *clean foreground*. Pada Gambar 6(g) yaitu *clean foreground* dua sepeda motor menunjukkan kualitas citra yang lebih baik dari proses pendeteksian *foreground* dengan menambahkan proses morfologi sebagai *removal noise*. Hal ini akan membuat hasil pendeteksian kendaraan sepeda motor lebih baik. Perbedaan Teknik blob analisis yang digunakan pada program ini dapat dijelaskan berdasarkan pemahaman terhadap Gambar 6(d) dan Gambar 6(h).

Pada Gambar 6(d) adalah *bounding boxes* menggunakan blob objek analisis. Dalam hal ini dilakukan sesuai luas area piksel kendaraan ringan yang sudah di tentukan sebelumnya. Pada Gambar 6(h) selanjutnya, menentukan *bounding boxes* sepeda motor. Hal ini dilakukan sesuai luas area piksel klasifikasi kendaraan yang sudah di tentukan sebelumnya. Sehingga pada frame video di atas dikenali sebagai 2 sepeda motor yang melewati area deteksi, dan akan bertambah jumlah kendaraan hasilnya pada layar sebesar 2 kendaraan sepeda motor.

Tahap selanjutnya yaitu, menentukan *bounding boxes* pada setiap komponen yang terhubung pada kendaraan dengan menggunakan blob objek analisis. Terakhir,



Gambar 7. Ilustrasi pengambilan gambar

menghitung jenis kendaraan berdasarkan klasifikasi kendaraan yaitu kendaraan berat (truk), kendaraan ringan, dan sepeda motor untuk dapat di gunakan sebagai input untuk menganalisa volume jumlah kendaraan dan jenis kendaraan. Kemudian proses terus berulang untuk seluruh *frame* dalam video.

D. Implementasi Sistem

Pengambilan gambar dilakukan menggunakan kamera digital. Pada penelitian ini pengambilan gambar dari kamera diwakili dengan file video. Gambar video yang diambil dari kamera selanjutnya disimpan ke dalam komputer dalam format .avi dengan ukuran 640x480 piksel.

Citra yang diteliti adalah citra jalan raya pada jalan Jalan T. Nyak Arief Jambotape Banda Aceh dengan durasi waktu 60 menit dan 50 frame per detik. Untuk memudahkan perhitungan sistem pengambilan video dibagi per 15 menit dalam durasi waktu 60 menit. Dimana 15 menit pertama disebut sequence 1; 15 menit kedua disebut sequence 2; 15 menit ketiga sequence 3; dan 15 menit keempat disebut sequence 4.

Pada Gambar 7. menunjukkan bahwa cara pengambilan gambar di lokasi penelitian. Yaitu kamera ditempatkan di atas jembatan penyeberangan jalan, kemudian menentukan panjang jalan yang dideteksi pada jalan raya.

IV. HASIL PENELITIAN

A. Hasil Pengujian

Model-model GMM terbentuk dari data warna piksel berdasarkan waktu. Model yang terbentuk dibagi menjadi 2 bagian, model *background* dan model *nonbackground*.

B. Hasil Klasifikasi Kendaraan

Berdasarkan hasil klasifikasi dapat dilihat bahwa, volume lalu lintas selama 1 jam pengamatan di lapangan untuk masing-masing jenis kendaraan adalah kendaraan berat sebesar 20 kendaraan, kendaraan ringan 999 kendaraan, sepeda motor 2293 kendaraan. Sedangkan

pada aplikasi didapat kendaraan berat 16 kendaraan, kendaraan ringan 1342 kendaraan, sepeda motor 2659 kendaraan. Untuk jumlah seluruh kendaraan selama 1 jam pengamatan di lapangan didapat volume lalu lintas sebesar 3312 kendaraan, dan menurut aplikasi didapat 3997 kendaraan. Tingkat error yang didapat adalah [11][12]:

Persentase error = $\{(3312-3997)/3312\} \times 100\% = 20,86\%$, maka tingkat akurasi pendeteksian kendaraan mencapai $= 100\% - 20,86\% = 79,32\%$.

Dari hasil perhitungan sistem masih belum sempurna, karena banyak sebab yang terjadi antaranya *noise* karena pengaruh bayangan, dalam hal ini adalah membedakan yang mana *foreground* dan *background*. Untuk membedakan *foreground* dan *background* maka diambil situasi yang bergerak. Situasi yang bergerak tersebut dianggap *foreground*, sedangkan bayangan itu mengikuti objek yang bergerak, sehingga bayangan pun dianggap *foreground*.

Oleh sebab itu untuk mengurangi *noise*, maka dilakukan beberapa hal yaitu: proses pengambilan video di waktu matahari seimbang, seperti sore hari atau pagi hari. Selanjutnya untuk mengurangi *noise* yang lain adalah menggunakan sistem yaitu menambahkan *clean foreground segmentasi*. Pada *clean foreground* ini menggunakan proses morfologi yaitu mengurangi angka-angka piksel yang kecil sehingga akan mengurangi *noise*. Jika ada dua kendaraan yang berdekatan, maka proses morfologi ini bisa memutus nilai bayangannya, sehingga jika putus maka itu di anggap bukan bagian dari kendaraan tersebut, karena dengan adanya proses morfologi ini nantinya tidak dianggap satu kendaraan tapi ada dua kendaraan yang terdeteksi.

V. KESIMPULAN

Telah dirancang piranti berbasis visi komputer untuk menentukan jumlah kendaraan berdasarkan jenis yaitu kendaraan berat, kendaraan ringan, dan sepeda motor, menggunakan metode *bounding box* yang diterapkan pada *Gaussian Mixture Models* (GMM). Pengklasifikasian kendaraan dilakukan berdasarkan ukuran *range* luas piksel objek kendaraan. Hasil pengujian aplikasi menunjukkan tingkat akurasi pendeteksian kendaraan mencapai 79,32%. Namun masih terdapat gangguan proses analisis disebabkan oleh berbagai kendala seperti bayangan kendaraan pada jalan, angin yang menyebabkan kamera bergoyang,

kondisi jalan yang tidak bersih, dan sebagainya. Sehingga mempengaruhi dalam menentukan luas area piksel kendaraan, karena bayangan tersebut berubah bentuk dan bergerak secara terus menerus, sehingga bayangan mengganggu *foreground* yang dihasilkan.

REFERENSI

- [1] Rizal Yulianto, dkk, "Pengukuran kepadatan arus lalu lintas menggunakan sensor kamera", Fakultas Teknik Elektro, Universitas Kristen Petra, [Online]. Available: <http://faculty.petra.ac.id/resmana/private/compvision/projects/NSSD/2003>.
- [2] Samuel Christian Susanto, dkk. (2010). "Perancangan pengaturan sisten traffict light dengan cctv dinamis: deteksi kepadatan jalan dengan citra digital pada maket jalan simpang empat." [Online]. Available: <http://repo.pens.ac.id/id/eprint/53>
- [3] Setiawan Hadi, dkk, "Deteksi objek kendaraan pada citra digital jalan raya menggunakan metode visi komputer", Jurnal Ilmiah Ilmu Komputer (JIHK) Vol : 8, Nomor : 2, 2012 : 215-223.
- [4] Eka Ardianto, dkk, "Implementasi metode image sbtracting dan metode regionprops untuk mendeteksi jumlah objek berwarna rgb pada file video", Jurnal Teknologi Informasi DINAMIK Volume 18, No.2, Juli 2013 : 91-100 ISSN : 0854-9524.
- [5] DIT.BSTP. "Laporan akhir perencanaan pengembangan IT untuk kepentingan lalu lintas perkotaan," Departemen Perhubungan Direktorat Jendral Perhubungan Darat Direktorat Bina Sistem Transportasi Perkotaan." Oktober 2009.
- [6] Helmiriawan, "Rancang bangun dan analisis sistem pemantau lalu lintas menggunakan opencv dengan algoritma canny dan blob detection", Skripsi, Departemen Teknik Elektro, Fakultas Teknik Universitas Indonesia, Depok, Juni 2011.
- [7] Bukhari dkk, "Rekayasa lalu lintas", Darussalam Banda Aceh, Indonesia : Bidang Studi Teknik transportasi Jurusan Teknik Sipil Fakultas Teknik Universitas Syiah Kuala, 1997.
- [8] Bukhari dkk, "Rekayasa lalu lintas", Darussalam Banda Aceh, Indonesia : Bidang Studi Teknik transportasi Jurusan Teknik Sipil Fakultas Teknik Universitas Syiah Kuala, 2002.
- [9] Fitroh Amaluddin, dkk, "Klasifikasi kendaraan menggunakan gaussian mixture model (GMM) dan fuzzy cluster means (FCM)", Jurnal EECCIS Vol. 9, No. 1, Juni 2015.
- [10] Dimas Wahyu Wibowo, dkk, "Perhitungan jumlah dan jenis kendaraan menggunakan metode fuzzy c-means dan segmentasi deteksi tepi canny", Jurnal EECCIS Vol. 7, No. 2, Desember 2013.
- [11] WikiHow. [Online]. Available: <http://id.m.wikihow.com/Menghitung-Galat-Persentase>
- [12] Math is Fun. [Online]. Available: <http://www.mathsisfun.com/numbers/percentage-error.html>